A Comparison between Grey Wolf Global Optimization Algorithm and two other Global Optimization Algorithms

عبدالباسط الهادي سعد السيليني
كلية التقنية الهندسية / جنزور
Abdulbasetsaad@gmail.com

مهند علي قشوط
كلية العلوم والتقنية / الرياينة
Mohanad86a@yahoo.com

جلال حسن الشيباني
المعهد العالي للتقنية الصناعية /انجيلة
jalalhiit@yahoo.com

اشرف ضو السملقي
المعهد العالي للعلوم والتقنية / طرابلس
ashrafdawo@gmail.com

Abstract:

In this paper a comprehension comparison between the Grey Wolf Global Optimization Algorithm (GWA) as a newly presented global optimization algorithm with two other wall-known algorithms including Cuckoo Search optimization algorithm (CSA), and Bat Optimization Algorithm (BA). All algorithms are applied on four complex benchmark functions. The purpose of this work is to identify the best algorithm in terms of converge speed and efficiency in finding the global optimum solution, where the converge speed is measured in terms the number of function evaluations. The simulation results show that the GWA algorithm with less function evaluations becomes first if the simulation time is important, while if efficiency is the significant issue, BA and CSA would have a better performance.

1. Introduction

Conventional mathematical based global optimization algorithms impose some difficulties on solving complex engineering problems which leads to development of alternative solutions such as stochastic based algorithms for searching near-global optimum solutions to problems.

Stochastic algorithms are random search methods that are mostly inspired from natural and social behavior of species. For example, a recently presented idea is based on the behavior of grey wolfs that track a social life [1], [2], also the biologic behavior of genes and the interaction of birds or frogs in a group can be key issues while inspiration.

In order to imitate the behavior of these species, which is guided by learning, adaptation, and evolution, various researchers have suggested computational systems to seek for solutions. The first stochastic -based method presented in the literature was the Genetic Algorithms [3]. the GAs method has been used in many complex applications in science and engineering [4]. However, GA suffered from drawbacks such as high computation time and easy getting stuck in local minima and become impractical and infeasible to use. On the other word, in industrial applications such as engineering and science [5], [6] two key issues that play the main role are the consumed time and the quality of the answer.

In an attempt to reduce computation time (CPU) and improve the quality of solutions other stochastic methods are suggested such as: GWA [7], CSA [8] and BA [9]. In this paper, three stochastic-based algorithms are reviewed with a special attention to the newly introduced GW algorithm and Performance comparison among the two algorithms is then presented. The paper is organized as follows. Section 2 is concerned with a review on GA Genetic Algorithms. Section 3 and 4 respectively focus on a brief review on GWA and CSA. Section 5 deals with elaborating on the bat behavior to implement the Bat algorithm (BA). Section

6 and 7 introduce the test objective functions and presents the simulation results and finally section 8 concludes the paper.

2. Genetic Algorithm (GA)

Genetic Algorithm (GA) was the first stochastic-based optimization technique developed by Jon Holland [10]. GA is a random search algorithm that is inspired by natural evolution. The algorithm starts with an initial set of points which are collectively known as population size. The algorithm has a fitness function that is used to calculate a function value of each point (candidate). The fitness value depends on how well the candidate solution solves problems and is the parameter that evaluates a candidate's rank in the movement towards the global optimal solution. One or two candidates are chosen from the population to perform a combination at each stage.

The recombination operations are of two types: crossover and mutation. In the first type, two candidates undergo crossover whereas, in mutation, only one candidate takes part. The crossover operation performs a randomized exchange between solutions, with the possibility of generating a better solution from a merely adequate one. This operation tends to narrow the search and move towards the global solution. On the other hand, mutation involves flipping possible solutions or an entity in a solution which expands the search exploration of the algorithm. Crossover and mutation rate are the probabilities at which the respective operations are performed. The choice of these probability values reflects the trade-off between exploration and exploitation (or convergence). A higher mutation rate, for example, leads to better exploration but can delay convergence. Moreover, a high crossover rate can lead to faster convergence but may get trapped in a local minimum.

GA has attracted the interest of many researchers as an effective approach to solve complex structure and achieve better performance. Croce et al. [11] presented a GA for solving job shop scheduling problems JSSPs with an encoding scheme that was based on preference rules. Sun et al. [12] developed a modified GA with a clonal selection and a life span strategy for the JSSPs; the developed algorithm was able to find 21 best known solutions out of 23 benchmarked instances. Lee and Yamak [13] proposed a GA with a new representation scheme that was based on operation completion time and its crossover was able to generate active schedules. Zhou et al. [14] developed a hybrid algorithm with a new representation scheme called random keys encoding. In this algorithm, GA was used to obtain an optimal schedule, and then a neighbor-hood search was introduced to perform local exploitation and increase the solution quality obtained from GA. Results showed that the hybrid framework performed better than GA and heuristic alone.

Typically, recombination gives an opportunity to reach new and better performing members who are then added to the population. Members in the population that have poor fitness values are thus gradually eliminated. This process is repeated until either a population member has the desired fitness value, hereby finding a solution, or the algorithm exceeds the time allocated to it, and is terminated.

In particular, GAs perform well for locating global optimization solutions - especially where the optimization problem is inexpensive. Furthermore, GA can be used in both unconstrained and constrained optimization problems. However, GA has a slow convergence speed even on the simple optimization problems and requires high computation time and a large number of function evaluations.

3. Grey Wolf Algorithm

The Grey Wolf Optimizer (GWO) approach is a recently proposed algorithm and is based on the behaviour of grey wolves in the wild [15]. GWO simulates the leadership policy and hunting strategy of a grey wolf's family in its natural habitat. GWO is similar to other nature-inspired population-based approaches such as GA, PSO and ACO. In a family of grey wolves, there are four different groups: alpha, beta, delta and omega. The alpha, which is always male, is in charge of making decisions in hunting, selecting rest and sleeping places, etc., and its decisions must be obeyed by the rest of the family. Due to its dominating role, alpha is placed at the top of the family pyramid. Beta is at the second level in the family, and its duty is to support the alpha's decisions or other family initiatives. Beta can be female or male, and, because of its experience working alongside the alpha, can replace the alpha if it becomes necessary. Beta acts as a counsellor to the alpha and ensures that the alpha's orders are applied in the community, while at the same time; it guides the lower-level wolves. Further down the pyramid is the Delta that must follow the orders of the alpha and beta wolves but has domination over the omega. The duty of the delta is to defend and provide safety to all family members. Omega is the lowest ranked among the grey wolf family and plays the role of scapegoat. Omegas are the last group of the grey wolf family to eat from the prey, and its duty is to take care of the new-born pups. These three groups are used to simulate the leadership hierarchy in the grey wolf family. The first three groups lead omega wolves to search the space. During this search, all members update their positions according to the locations of the alpha, beta and delta.

In GWO, there are three steps of hunting which must be realized: finding prey, surrounding prey, and finally attacking and killing prey. Through the optimization procedure, the three most effective candidate solutions are alpha, beta and delta, as they are likely to be at the location of the optimal solution. Meanwhile the omega wolves must relocate with respect to the location of the other groups. As laid out in the GWO algorithm, alpha is the fittest candidate, but beta and delta gain better information about the potential position of prey than omegas. Accordingly, the best three solutions are saved in the database, while the rest of the search agents (omegas) are obliged to update their position according to the position of the best solution so far. In GWA, n is the wolf population, k indicates the number of iteration, A and C are random parameters $A = (1,0), C = (1,1)$, x_p represents the location vector of the prey, x is the location of the agent, and a is a random value to update position.

The GW algorithm is recognized as being a capable and efficient optimization tool that can provide a very accurate result without becoming trapped in local optima [16]. Because of its inherent advantages, GWA is used in several optimal design applications. El-Fergany and Hasanien [17] integrated GWO and DE to handle single and complex power flow problems. Zawbaa et al. [18] developed and applied a new version of GWO called the binary grey wolf optimization (BGWO) to find the optimal zone of the complex design space. Kohli and Arora [19] introduced the chaos theory into the GWO algorithm (CGWO) with the aim of accelerating its global convergence speed. Mittal et al. [20]

proposed a modified grey wolf optimizer (MGWO) to improve the exploration and exploitation capability of the GWO that led to optimal efficiency of the method.

4. Cuckoo Search Algorithm

Another recent nature-inspired global optimization method is the Cuckoo Search (CS) approach developed by Yang in 2009 [21]. The CS method is based on the natural obligatory brood parasitic behaviour of cuckoo birds in integration with the Lévy flight. A cuckoo bird places its eggs in another bird’s nest to be brooded by the mother bird of another species. In some cases, other birds engage in battle with the stranger cuckoos when the other bird realizes that the eggs in her nest are not her own. In this case, the other bird either destroys the unwelcome eggs in the nest or leaves its own nest and rebuilds a new one elsewhere. Some cuckoo female species have developed a new strategy based on imitating the colours and shapes of the eggs of other birds to increase the chance of reproduction and decrease the probability of desertion by the other bird. In general, the cuckoo’s eggs hatch before the other bird’s eggs, thus the first job of the cuckoo chick is to get rid of the other bird’s eggs to increase its own chance of being fed by the resident mother bird. This knowledge of the Cuckoo bird has been used to develop the CS algorithm.

The easiest way of applying the CS algorithm is achieved through the following three steps [22]. First, every cuckoo bird places only one egg at a time in a random nest. Second, the best nests (solutions) with a good quality of eggs are selected for the next population. Third, the number of nests is constant, and the egg deposited by a cuckoo is recognized by the other bird with a probability of $P_a \in [0,1]$. Therefore, the other bird may either destroy the alien eggs or relinquish the nest and establish a new nest. The last assumption can be estimated as the fraction P_a of the n nests when new nests (completely new solutions) are substituted. In the CS method, every egg in a nest expresses a solution, and each cuckoo can only deposit one egg. This algorithm can also be used when the problem is more complex such as where each nest could hold several eggs representing a number of solutions. Further, each cuckoo can be simply considered as a random point in the design space while the nest is the memories that are used to keep the previous solutions and compare them with the next solutions.

CS is used in dealing with high-dimensional, linear and nonlinear GO problems. A recent study showed that CS is more effective and robust than PSO and GA in multi-modal objective functions [22]. This is partially due to that there are only a limited number of parameters to adjust in the CS method compared to other GO algorithms such as PSO and GA. A comprehensive description of the structure of the CS method is available in [22].

Since the development of the CS method in 2010, several studies have been introduced to improve its performance. Walton et al. [23] modified the CS algorithm to be more effective in handling nonlinear GO problems such as mesh generation. Yildiz [24] employed the CS algorithm to find the optimal parameters for a machine in the milling process. Vazquez [25] used the CS algorithm with artificial neural network model can to deal with different linear and non-linear problems. Kaveh and Bakhshpoori [26] applied the CS algorithm in designing steel frames. Speed [27] modified the CS algorithm to be used efficiently in dealing with large-scale problems.

5. Bat Algorithm Method

The Bat Algorithm (BA) is a mature nature-based algorithm proposed by Yang based on prey tracking behaviour [28]. Using the concept of echolocation, bats create sounds while flying about hunting for food. These sounds are reflected to the bat providing useful information about the targets. This mechanism enables bats to identify the type of the objects, the distance from the target, and the kind and speed of the prey. Bats have the capability to establish three-dimensional pictures around the hunting area using their advanced echolocation strategy.

While hunting, bats move randomly with velocity v_i at location x_i sending pulses with a range of frequency $f \in [f_{min}, f_{max}]$ (wavelength of λ and loudness A_0) while searching for prey. Bats can control the frequency pulses and regulate the rate of pulse emissions $r \in [0,1]$ where 0 expresses that there are no emissions, and 1 expresses that the emissions of bats are at their maximum power. In BA, the loudness can be controlled from maximum (positive) A_0 to the lowest value. Note that $A_0 = 0$ expresses that a bat has reached its target and has stopped releasing any sounds. In BA, x^* is the best global solution at present and is obtained from among all achieved solutions. The value of f depends on the size of the design space. The higher the frequency, the shorter the wavelength and the shorter the travelled distance. Bats use a limited range of frequencies from 200 to 500 kHz. The steps of BA are described in [29].

BA is known as a very robust and efficient method in dealing with many engineering optimization problems [30]. Although many publications on this algorithm exist, BA still attracts a great deal of interest from researchers in a wide range of applications.

Many researchers studied BA to ensure that it is able to avoid becoming trapped into local minima. For instance, Xie et al. [31] introduced a combination of Lévy flight with the BA (DLBA). Lin et al. [32] presented another hybrid of Lévy flight and bat approach (CLBA) for parameter approximation in a nonlinear dynamic model. Yılmaz and Kucuksille [33] motivated by the PSO algorithm and the ABC algorithm, proposed an improved bat algorithm (IBA) to advance the exploration mechanism of the algorithm. Wang and Guo [34] integrated the harmony search (HS) method into BA, and developed a hybrid metaheuristic (HSBA) method to increase the convergence speed of BA. Zhu et al. [35] improved the exploration capability of BA by modifying its equations. Gandomi and Yang [36] replaced the four parameters in BA by different chaotic systems to increase the global search capability of BA. Kielkowicz and Grela [37] introduced some modification to the Bat Algorithm to solve nonlinear engineering optimization problems.

6. Simulation Results

All the stochastic-based algorithms described above are coded in Matlab® and the simulations is carried out on a 2.2 GHz CORE I5 Laptop. The performance of the three stochastic-based algorithms is compared using four benchmark problems whose description is given in the following table and figure 1.

No.	Fun.	D	Search Space	Analytic *f**	Fun. Category
f1	Ackley	5	[-32, 32]	0.0000	M-Modal
f2	Griewank	6	[-100, 100]	0.0000	M-Modal
f3	Alpine	8	[-10, 10]	0.0000	M-Modal
f4	Egg Carte	10	[-5, 5]	0.0000	M-Modal

Table 1 Tested Functions [38]

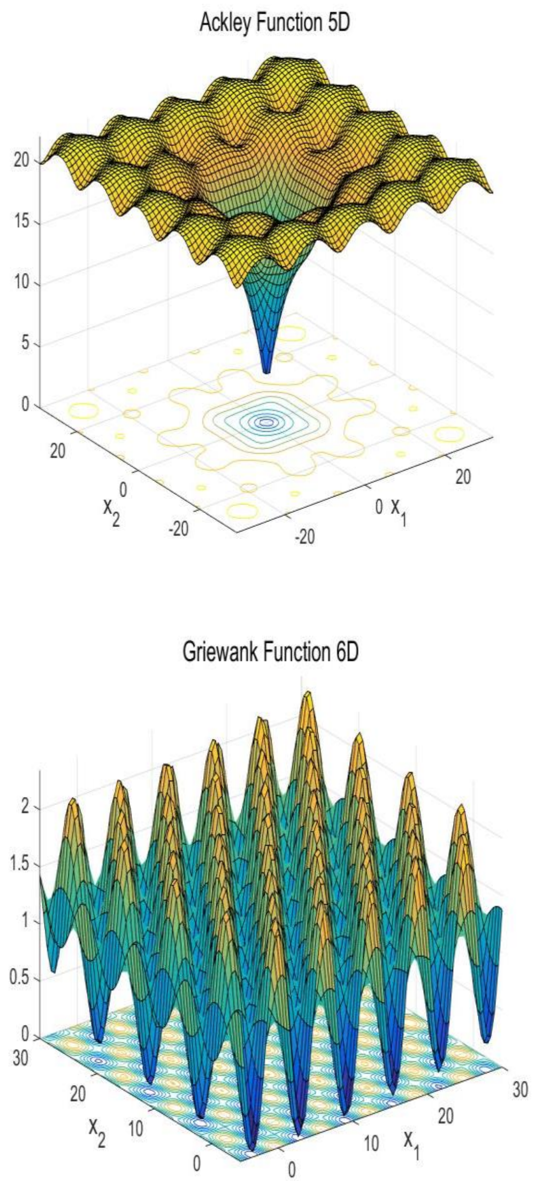

Figure 1. Samples of Tested Multimodal Functions

7. Results and Discussion

The convergence speed for the algorithms are demonstrated for *f1* function in figures 2 and the Table 2 includes the corresponding minimum values of the tested functions and the number of function evaluations NFE for the algorithms carried out on tested functions. Similarly figures 3 illustrate the convergence speed for the algorithms when applied to solve *f4*. it can be seen that the convergence speed of GW algorithm is better than the other two algorithms (BA, CSA) on the entire set considered. The study concludes that GW algorithm requires a significantly lesser number of function evaluations to converge with more accuracy than needed for the BA and CSA for all chosen problems.

Table 2 Qualitative Results for The Tested

Fun.	***f1***		***f2***		***f3***		***f4***	
	Obt. f*	***NFE***	***Obt. f****	***NFE***	***Obt. f****	***NFE***	***Obt. f****	***NFE***
GWA	6.99 E-7	359	3.89 E-6	402	1.89 E-8	471	3.68 E-6	421
BA	1.59 E-5	412	1.64 E-4	465	3.80 E-4	481	3.95E-5	498
CSA	1.73 E-3	488	2.36 E-3	491	1.84 E-4	479	5.96 E-5	467

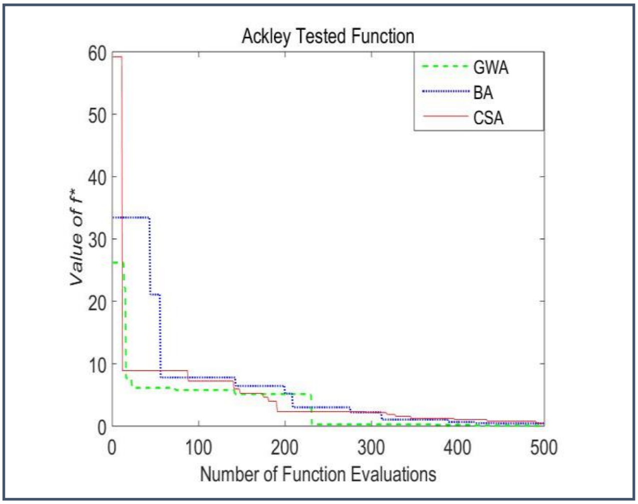

Figure 2. Convergence speed for *f1*

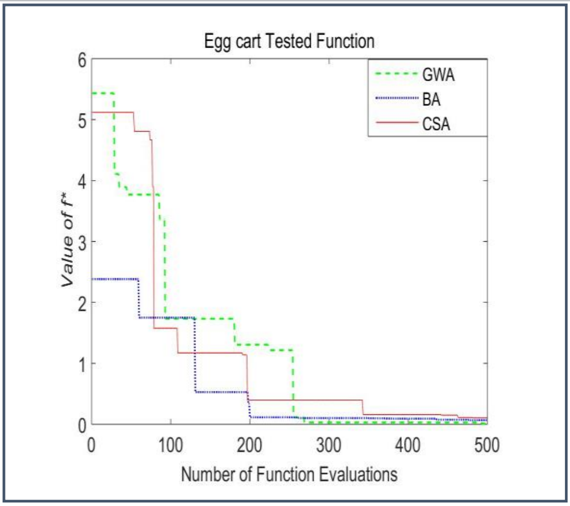

Figure 3. Convergence speed for *f7*

8. Conclusion

Achieving the goals basically depend on two key issues, less number of function evaluations (NFE), or efficiency in approaching to the answer. For instance, in some fields such as missile control, both of those issues are equally important and inseparable from each other. Accordingly, on the basis of figures it is quite obvious that if speed (less function evaluation) is the more important criterion, GW algorithm is the most recommended one among the compared ones here. However, if the accuracy in reaching to the specific point is of greater importance, it would be suggested to choose between BA or CSA algorithm.

References

[1] Kamboj, V.K., S.K. Bath, and J.S. Dhillon, Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer. Neural Computing and Applications, 2016. 27(5): p. 1301-1316.
[2] Garg, H., A hybrid PSO-GA algorithm for constrained optimization problems. Applied Mathematics and Computation, 2016. 274: p. 292-305.
[3] Samora, I., et al., Simulated annealing in optimization of energy production in a water supply network. Water resources management, 2016. 30(4): p. 1533-1547.
[4] Nik, M.A., et al., A comparative study of metamodeling methods for the design optimization of variable stiffness composites. Composite Structures, 2014. 107: p. 494-501.
[5] Yao, X., A new simulated annealing algorithm. International Journal of Computer Mathematics, 1995. 56(3-4): p. 161-168.
[6] Xinchao, Z., Simulated annealing algorithm with adaptive neighborhood. Applied Soft Computing, 2011. 11(2): p. 1827-1836.
[7] Yusof, Y. and Z. Mustaffa, Time Series Forecasting of Energy Commodity using Grey Wolf Optimizer. Vol. 1. 2015. 25-30.
[8] Yatim, J., A. Zain, and N.E.N. Bazin, Cuckoo Search Algorithm for Optimization Problems—A Literature Review and its Applications. Vol. 28. 2014.
[9] Yang, X.-S., A New Metaheuristic Bat-Inspired Algorithm. Vol. 284. 2010.
[10] Yusof, R., et al., Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm. Applied soft computing, 2011. 11(8): p. 5782-5792.
[11] Della Croce, F., R. Tadei, and G. Volta, A genetic algorithm for the job shop problem. Computers & Operations Research, 1995. 22(1): p. 15-24.

[12] Sun, L., X. Cheng, and Y. Liang, Solving job shop scheduling problem using genetic algorithm with penalty function. International Journal of Intelligent information processing, 2010. 1(2): p. 65-77.
[13] Lee, K.-M., T. Yamakawa, and K.-M. Lee. A genetic algorithm for general machine scheduling problems. in Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES'98. 1998 Second International Conference on. 1998. IEEE.
[14] Liu, L. and Y. Xi. A hybrid genetic algorithm for job shop scheduling problem to minimize makespan. in Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on. 2006. IEEE.
[15] Mirjalili, S., S.M. Mirjalili, and A. Lewis, Grey Wolf Optimizer. Advances in Engineering Software, 2014. 69: p. 46-61.
[16] Precup, R.-E., et al., An Easily Understandable Grey Wolf Optimizer and Its Application to Fuzzy Controller Tuning. Algorithms, 2017. 10(2): p. 68.
[17] El-Fergany, A. and H. Hasanien, Single and Multi-objective Optimal Power Flow Using Grey Wolf Optimizer and Differential Evolution Algorithms. Vol. 43. 2015. 1548-1559.
[18] Zawbaa, H., E. Emary, and A.E. Hassanien, Binary Grey Wolf Optimization Approaches for Feature Selection. Vol. 172. 2016. 371-381.
[19] Kohli, M. and S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems. 2017.
[20] Mittal, N., U. Singh, and B. Singh Sohi, Modified Grey Wolf Optimizer for Global Engineering Optimization. Vol. 2016. 2016. 1-16.
[21] Yang, X.-S. and S. Deb, Cuckoo search: recent advances and applications. Neural Computing and Applications. Vol. 24. 2014.
[22] Yatim, J., A. Zain, and N.E.N. Bazin, Cuckoo Search Algorithm for Optimization Problems—A Literature Review and its Applications. Vol. 28. 2014.

[23] Walton, S., et al., Modified cuckoo search: A new gradient free optimisation algorithm. Chaos, Solitons & Fractals, 2011. 44(9): p. 710-718.
[24] Yildiz, A., Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. Vol. 64. 2012.
[25] Vazquez, R.A., Training spiking neural models using cuckoo search algorithm. IEEE Congress on Eovlutionary Computation (CEC'11, 2011(2011): p. 679-686.
[26] Kaveh, A. and T. Bakhshpoori, Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights. The Structural Design of Tall and Special Buildings, 2013. 22(13): p. 1023-1036.
[27] Speed, E.R., Evolving a Mario agent using cuckoo search and softmax heuristics. 2010. 1-7.
[28] Yang, X.-S., A New Metaheuristic Bat-Inspired Algorithm. Vol. 284. 2010.
[29] Afrabandpey, H., et al., A Novel Bat Algorithm Based on Chaos for Optimization Tasks. 2014.
[30] Al-Betar, M., et al., Bat-inspired Algorithms with Natural Selection mechanisms for Global optimization. 2017.
[31] Xie, J., Y.-Q. Zhou, and H. Chen, A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory. Vol. 2013. 2013. 453812.
[32] Lin, J.H., et al., A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. Vol. 2. 2012. 56-63.
[33] Yılmaz, S. and E.U. Kucuksille, Improved Bat Algorithm (IBA) on Continuous Optimization Problems. Vol. 1. 2013.
[34] Wang, G. and L. Guo, A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization. Journal of Applied Mathematics, 2013. 2013: p. 21.

[35] Zhu, B., et al., A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization. Vol. 2016. 2016. 1-17.
[36] Gandomi, A. and X.-S. Yang, Chaotic bat algorithm. Vol. 5. 2013.
[37] Kielkowicz, K. and D. Grela, Modified Bat Algorithm for Nonlinear Optimization. International Journal of Computer Science and Network Security (IJCSNS), 2016. 16(10): p. 46.
[38] Asadzadeh, L. and K. Zamanifar, An agent-based parallel approach for the job shop scheduling problem with genetic algorithms. Mathematical and Computer Modelling, 2010. 52(11): p. 1957-1965.